

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims

1. (Previously Presented) A computerized method for enabling access to a plurality of service engines, wherein each service engine enables access to service resources, comprising:
providing a plurality of service class implementations for service engines from different vendors, wherein each service class implementation provides an implementation of methods and objects from a same abstract service class;

instantiating a service object for one service engine in response to at least one called method from one of the service class implementations, wherein the service object includes information on the service engine;

receiving a method call from one service class implementation requesting information on service engine resources for one named service; and

using the service object to access the requested information to return to the method call.

2. (Previously Presented) The method of claim 1, wherein the method call is received from at least one application program including methods from at least one of the service class implementations to access information on at least one service from the service objects.

3. (Previously Presented) The method of claim 1, wherein the service engines comprise workflow products from different vendors, wherein the workflow products comprise computer programs enabling implementation of a computer implemented workflow defining a series of processes to be performed by users at computers with respect to a computer implemented work item.

4. (Original) The method of claim 3, wherein the workflow service class implementations from different vendors each includes methods and objects from a same abstract workflow service class specifying methods and objects to include in all workflow service class implementations.

5. (Original) The method of claim 4, wherein the abstract workflow service class specifies a method to determine a data store used by resources in the service, wherein the workflow service class implementations for the workflow services implement a method to determine from the service object for the service the data store used by the service.

6. (Original) The method of claim 4, wherein the abstract workflow service class specifies methods to access information on workflows, workflow templates, and worklists.

7. (Original) The method of claim 4, wherein the abstract workflow class specifies a method for a user to connect to the workflow service by passing user information that is used to authenticate the user to access the workflow service, and wherein the workflow service class implementations include methods to enable a connection between a user and one workflow service.

8. (Original) The method of claim 7, further comprising:
receiving a method call for the user to connect specifying a user name, authentication information, and one service;
instantiating a connection object if authentication passed including information on the specified user name, authentication information, and the service engine that can be accessed with the authentication; and
returning a handle to the connection object for use in obtaining authentication information to access the service engine specified in the connection object.

9. (Original) The method of claim 8, wherein the connection object is stored in one service object.

10. (Currently Amended) The method of claim 1, further comprising:
receiving one handle to one connection object; and
returning [[the]] authentication information from the connection object for use in accessing the service resources.

11. (Previously Presented) The method of claim 1, wherein the service class implementations for the service engines include both a client side service class and server side service class, wherein the client side service class includes methods and objects used to access information on resources available at one service engine, and wherein the server side service class includes methods and objects to access information on resources available at one service engine, wherein the client side service class methods are invoked on a client system and the server side service class methods are invoked on a server including the service engine.

12. (Original) The method of claim 11, wherein a call to one method in the client side service class implementation on the client transfers the called method to the server, wherein the called method is executed on the server.

13. (Original) The method of claim 1, wherein information on the service engine is maintained in the service object and wherein using the service object to access the requested information further comprises:

if the method call requests information in the service object, returning the requested information from the service object.

14. (Original) The method of claim 13, further comprising:
if the method call requests information not included in the service object, executing the method to access the requested information from the service engine.

15. (Original) The method of claim 13, further comprising:
accessing authentication from the service object;
providing the accessed authentication information to the service engine;
using, with the service engine, the accessed authentication information to authenticate access, wherein the service engine executes the method to access the requested information if access is authenticated.

16. (Original) The method of claim 14, wherein accessing the requested information from the service engine further comprises:

translating the method to native code capable of being executed by the service engine.

17. (Previously Presented) A system for enabling access to a plurality of service engines, wherein each service engine enables access to service resources, comprising:

at least one computer readable medium including a plurality of service class implementations for service engines from different vendors, wherein each service class implementation provides an implementation of methods and objects from a same abstract service class;

means for instantiating a service object in one compute readable medium for one service engine in response to at least one called method from one of the service class implementations, wherein the service object includes information on the service engine;

means for receiving a method call from one service class implementation requesting information on service engine resources for one named service; and

means for using the service object to access the requested information to return to the method call.

18. (Previously Presented) The system of claim 17, wherein the method call is received from at least one application program including methods from at least one of the service class implementations to access information on at least one service from the service objects.

19. (Previously Presented) The system of claim 17, wherein the service engines comprise workflow products from different vendors, and wherein the at least one computer readable medium includes workflow service class implementations from different vendors, wherein the workflow products comprise computer programs enabling implementation of a computer implemented workflow defining a series of processes to be performed by users at computers with respect to a computer implemented work item.

20. (Currently Amended) The system of claim ~~[[17]]~~ 19, wherein the workflow service class implementations from different vendors each includes methods and objects from a

same abstract workflow service class specifying methods and objects to include in all workflow service class implementations.

21. (Original) The system of claim 20, wherein the abstract workflow service class specifies a method to determine a data store used by resources in the service, wherein the workflow service class implementations for the workflow services implement a method to determine from the service object for the service the data store used by the service.

22. (Original) The system of claim 20, wherein the abstract workflow service class specifies methods to access information on workflows, workflow templates, and worklists.

23. (Original) The system of claim 20, wherein the abstract workflow class specifies a method for a user to connect to the workflow service by passing user information that is used to authenticate the user to access the workflow service, and wherein the workflow service class implementations include methods to enable a connection between a user and one workflow service.

24. (Original) The system of claim 23, further comprising:
means for receiving a method call for the user to connect specifying a user name, authentication information, and one service;
means for instantiating a connection object in one computer readable medium if authentication passed, wherein the connection object includes information on the specified user name, authentication information, and the service engine that can be accessed with the authentication; and
means for returning a handle to the connection object for use in obtaining authentication information to access the service engine specified in the connection object.

25. (Original) The system of claim 24, wherein the connection object is stored in one service object in the computer readable medium.

26. (Currently Amended) The system of claim 17, further comprising:
means for receiving one handle to one connection object; and
means for returning [[the]] authentication information from the connection object for use
in accessing the service resources.

27. (Previously Presented) The system of claim 17, wherein the service class
implementations for the service engines include both a client side service class and server side
service class, wherein the at least one computer readable medium includes a client computer
readable medium and a server computer readable medium, further comprising:

(a) a client system, including:

(i) the client computer readable medium, wherein the client computer readable
medium includes the client side service class, and wherein the client side service class
includes methods and objects used to access information on resources available at one
service engine;

(ii) means for executing the client side service class;

(b) a server system, including:

(i) the server computer readable medium, wherein the server computer readable
medium includes the server side service class, and wherein the server side service class
includes methods and objects to access information on resources available at one service
engine; and

(ii) means for executing the server side service class.

28. (Original) The system of claim 27, further comprising:
means for transferring a call to one method in the client side service class implementation
on the client to the server, wherein the called method is executed on the server.

29. (Original) The system of claim 17, wherein information on the service engine is
maintained in the service object and wherein using the service object to access the requested
information further comprises:

means for returning the requested information from the service object if the method call
requests information in the service object.

30. (Original) The system of claim 29, further comprising:
means for executing the method to access the requested information from the service engine if the method call requests information not included in the service object.

31. (Original) The system of claim 29, further comprising:
means for accessing authentication from the service object; and
means for providing the accessed authentication information to the service engine,
wherein the service engine uses the accessed authentication information to authenticate access,
wherein the service engine executes the method to access the requested information if access is authenticated.

32. (Original) The system of claim 30, wherein the means for executing the method to access the requested information from the service engine further performs:
translating the method to native code capable of being executed by the service engine.

33. (Previously Presented) An article of manufacture including code for enabling access to a plurality of service engines, wherein each service engine enables access to service resources, wherein the code is capable of causing operations to be performed, the operations comprising:
providing a plurality of service class implementations for service engines from different vendors, wherein each service class implementation provides an implementation of methods and objects from a same abstract service class;
instantiating a service object for one service engine in response to at least one called method from one of the service class implementations, wherein the service object includes information on the service engine;
receiving a method from one service class implementation requesting information on service engine resources for one named service; and
using the service object to access the requested information to return to a method call.

34. (Previously Presented) The article of manufacture of claim 33, wherein the method call is received from at least one application program including methods from at least

one of the service class implementations to access information on at least one service from the service objects.

35. (Previously Presented) The article of manufacture of claim 33, wherein the service engines comprise workflow products from different vendors, wherein the workflow products comprise computer programs enabling implementation of a computer implemented workflow defining a series of processes to be performed by users at computers with respect to a computer implemented work item.

36. (Original) The article of manufacture of claim 35, wherein the workflow service class implementations from different vendors each includes methods and objects from a same abstract workflow service class specifying methods and objects to include in all workflow service class implementations.

37. (Original) The article of manufacture of claim 36, wherein the abstract workflow service class specifies a method to determine a data store used by resources in the service, wherein the workflow service class implementations for the workflow services implement a method to determine from the service object for the service the data store used by the service.

38. (Original) The article of manufacture of claim 36, wherein the abstract workflow service class specifies methods to access information on workflows, workflow templates, and worklists.

39. (Original) The article of manufacture of claim 36, wherein the abstract workflow class specifies a method for a user to connect to the workflow service by passing user information that is used to authenticate the user to access the workflow service, and wherein the workflow service class implementations include methods to enable a connection between a user and one workflow service.

40. (Original) The article of manufacture of claim 39, further comprising:
receiving a method call for the user to connect specifying a user name, authentication information, and one service;
instantiating a connection object if authentication passed including information on the specified user name, authentication information, and the service engine that can be accessed with the authentication; and
returning a handle to the connection object for use in obtaining authentication information to access the service engine specified in the connection object.

41. (Original) The article of manufacture of claim 40, wherein the connection object is stored in one service object.

42. (Currently Amended) The article of manufacture of claim 33, further comprising:
receiving one handle to one connection object; and
returning [[the]] authentication information from the connection object for use in accessing the service resources.

43. (Previously Presented) The article of manufacture of claim 33, wherein the service class implementations for the service engines include both a client side service class and server side service class, wherein the client side service class includes methods and objects used to access information on resources available at one service engine, and wherein the server side service class includes methods and objects to access information on resources available at one service engine, wherein the client side service class methods are invoked on a client system and the server side service class methods are invoked on a server including the service engine.

44. (Original) The article of manufacture of claim 43, wherein a call to one method in the client side service class implementation on the client transfers the called method to the server, wherein the called method is executed on the server.

45. (Original) The article of manufacture of claim 33, wherein information on the service engine is maintained in the service object and wherein using the service object to access the requested information further comprises:

if the method call requests information in the service object, returning the requested information from the service object.

46. (Original) The article of manufacture of claim 45, further comprising:
if the method call requests information not included in the service object, executing the method to access the requested information from the service engine.

47. (Original) The article of manufacture of claim 45, further comprising:
accessing authentication from the service object;
providing the accessed authentication information to the service engine; and
using, with the service engine, the accessed authentication information to authenticate access, wherein the service engine executes the method to access the requested information if access is authenticated.

48. (Original) The article of manufacture of claim 46, wherein accessing the requested information from the service engine further comprises:
translating the method to native code capable of being executed by the service engine.

49. (New) The method of claim 11, further comprising:
receiving, at the client system, a construct method to construct a client side service object to access the service engine;
constructing the client side service object in response to the construct method;
communicating the construct method to the server to cause the server to construct a server side service object; and
issuing method calls from client applications in the client system to access information from the service engine using the constructed client-side and server side service objects.

50. (New) The system of claim 27,
wherein the means for executing the client side service class at the client system is
enabled to perform:
receiving a construct method to construct a client side service object to access the service
engine;
constructing the client side service object in response to the construct method;
communicating the construct method to the server to cause the server to construct a
server side service object; and
issuing method calls from client applications in the client system to access information
from the service engine using the constructed client-side and server side service objects.

51. (New) The article of manufacture of claim 43, wherein the operations further
comprise:
receiving, at the client system, a construct method to construct a client side service object
to access the service engine;
constructing the client side service object in response to the construct method;
communicating the construct method to the server to cause the server to construct a
server side service object; and
issuing method calls from client applications in the client system to access information
from the service engine using the constructed client-side and server side service objects.